

Strange Adventure

João Paulo Souza Ranieri

Prof.^a Sílvia F. M. Brandão, Ms

Roteiro

- Objetivos
- Resumo
- Requisitos
- Metodologia de Desenvolvimento
- Ambiente de Desenvolvimento
- Testes
- Implantação

Objetivos

- Apresentar as ferramentas utilizadas no desenvolvimento de um jogo.
- Demonstrar o desenvolvimento do jogo.

Resumo

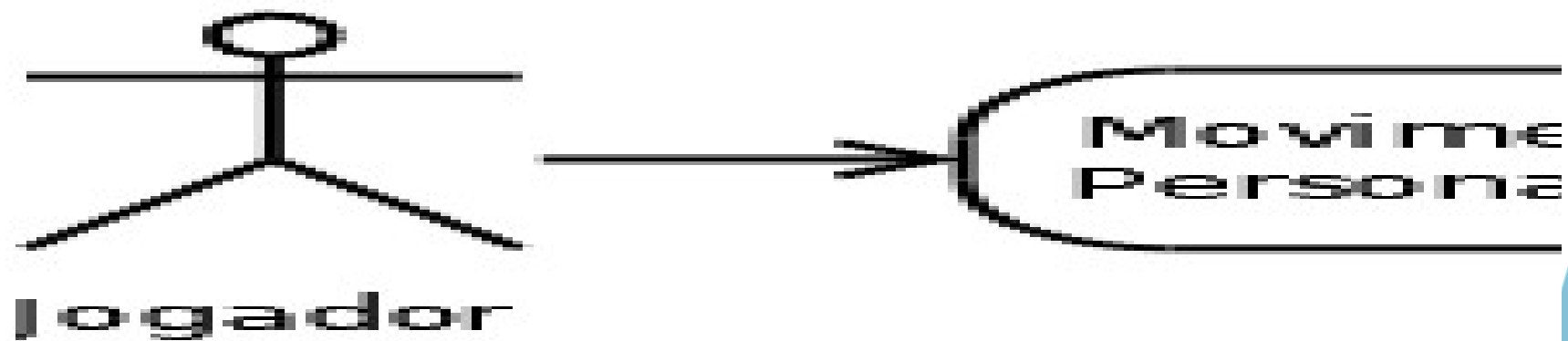
- O software Strange Adventure se trata de um jogo de aventura, desenvolvido na ferramenta Unity.

Requisitos

- Movimentação.
- Colisão com Objetos.
- Interação com Objetos.
- Interação com Inimigos.
- Sistema de Vida e Morte.

Caso de Uso

Movimentação



Caso de Uso

Colisão com Inimigos e Objetos



Metodologia de Desenvolvimento

- Metodologia Evolutiva.
- Desenvolvimento de um protótipo.
- Teste.
- Evolução do protótipo.

Ambiente de Desenvolvimento

Hospedagem



Ferramentas



Imagens do Ambiente de Desenvolvimento

Declaração de Variáveis

```
private Animator playerAnimator;  
private Rigidbody2D playerRb;  
public Transform groundCheck; //objeto responsavel por detectar se o personagem esta sobre uma superficie  
public Collider2D standing, crouching; //colisor em pe e agachado  
  
public bool Grounded; //indidca se o personagem esta pisando no chao  
public bool lookLeft; //indica se o personagem esta virada para esquerda  
  
public int idAnimation; //indica o id da animacao  
  
private float horizontal, vertical;  
public float speed; //velocidade de movimento do personagem  
public float jumpForce; //forca aplicada para gerar o pulo do personagem
```

Função de Virar Personagem

```
//funcao para virar o personagem
void flip(){
    lookLeft = !lookLeft; //invert o valor da variavel booleana
    float x = transform.localScale.x;
    x *= -1; //invert o valor do scale x
    transform.localScale = new Vector3(x, transform.localScale.y, transform.localScale.z);
}
```

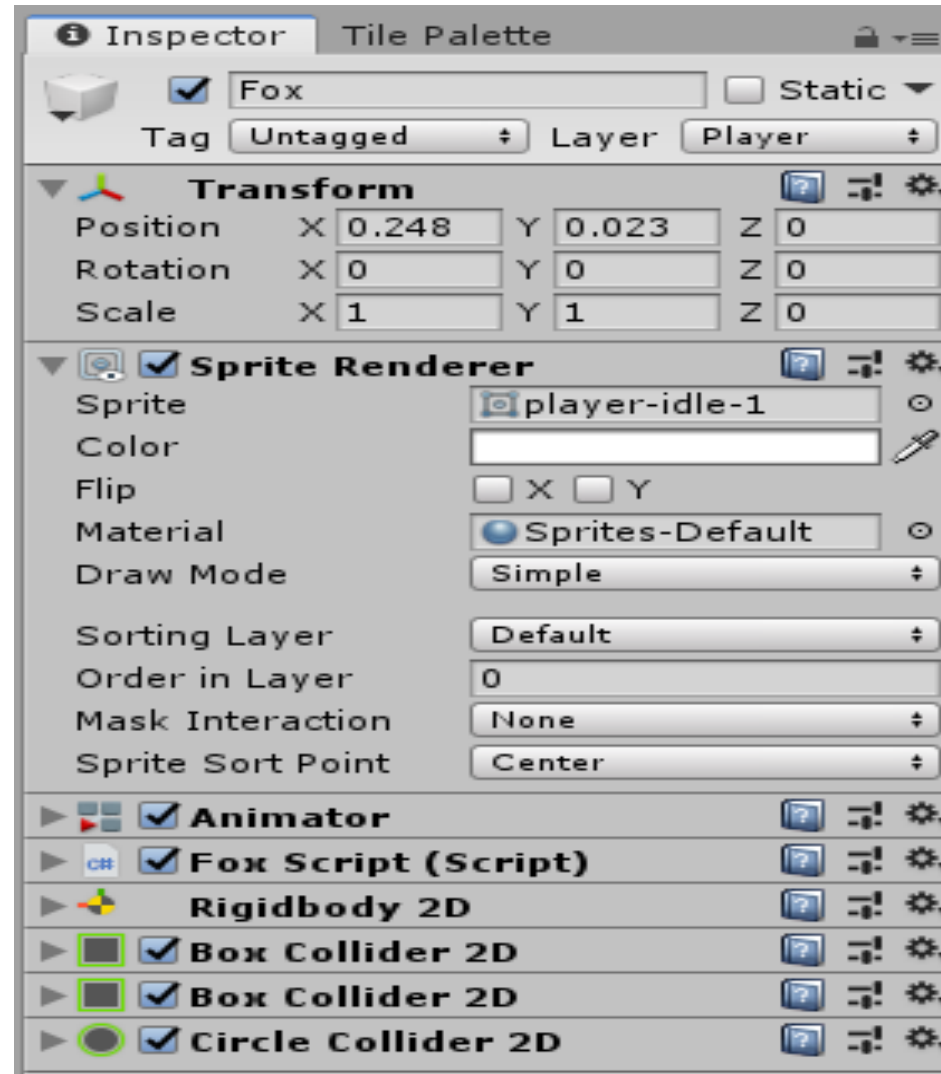
Função de Patrulha

```
protected void patrol(){
    Debug.DrawRay(transform.position, dir * -1 * distanceChangeRoute, Color.red);
    RaycastHit2D hit = Physics2D.Raycast(transform.position, dir * -1, distanceChangeRoute, layerObstacles);

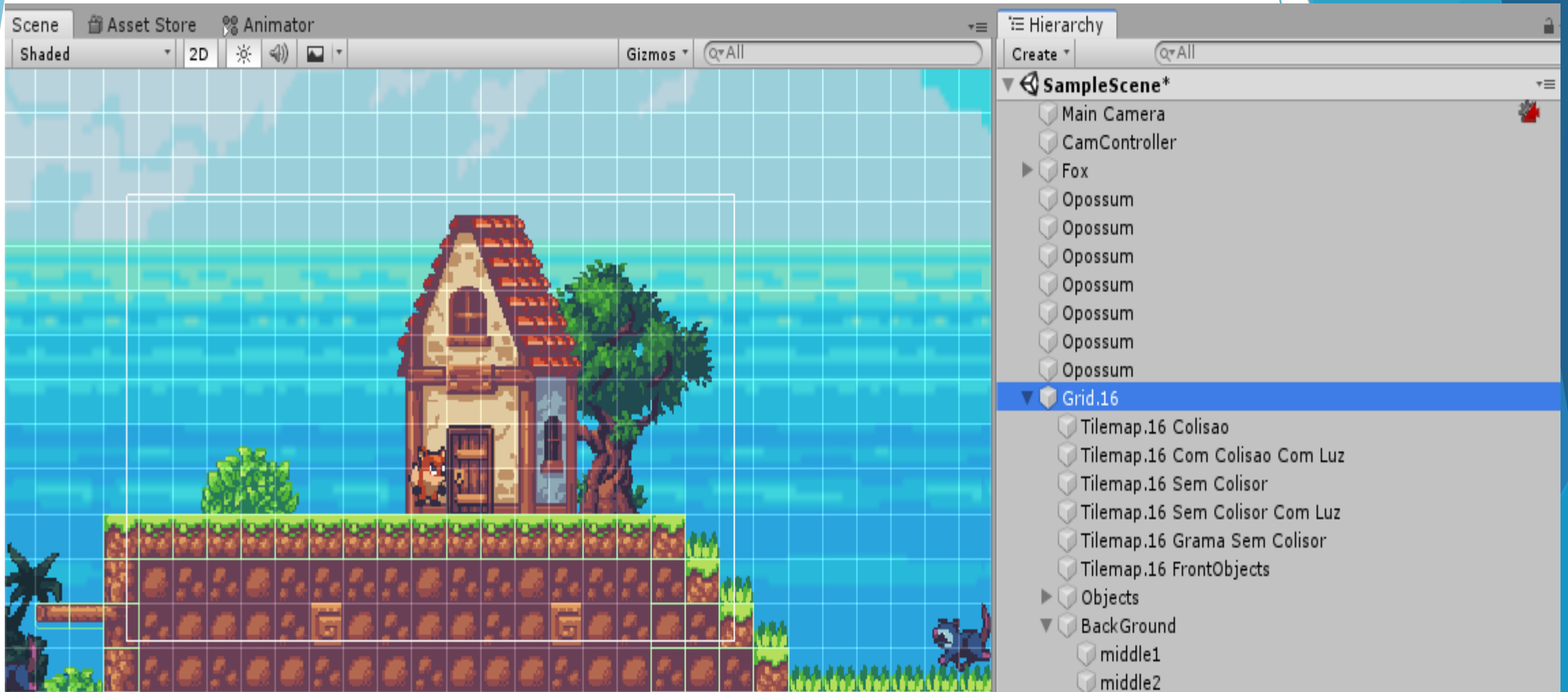
    if (hit) {
        flip();
        dir *= -1;
    }

    enemyRB.velocity = new Vector2(speed, enemyRB.velocity.y);
}
```

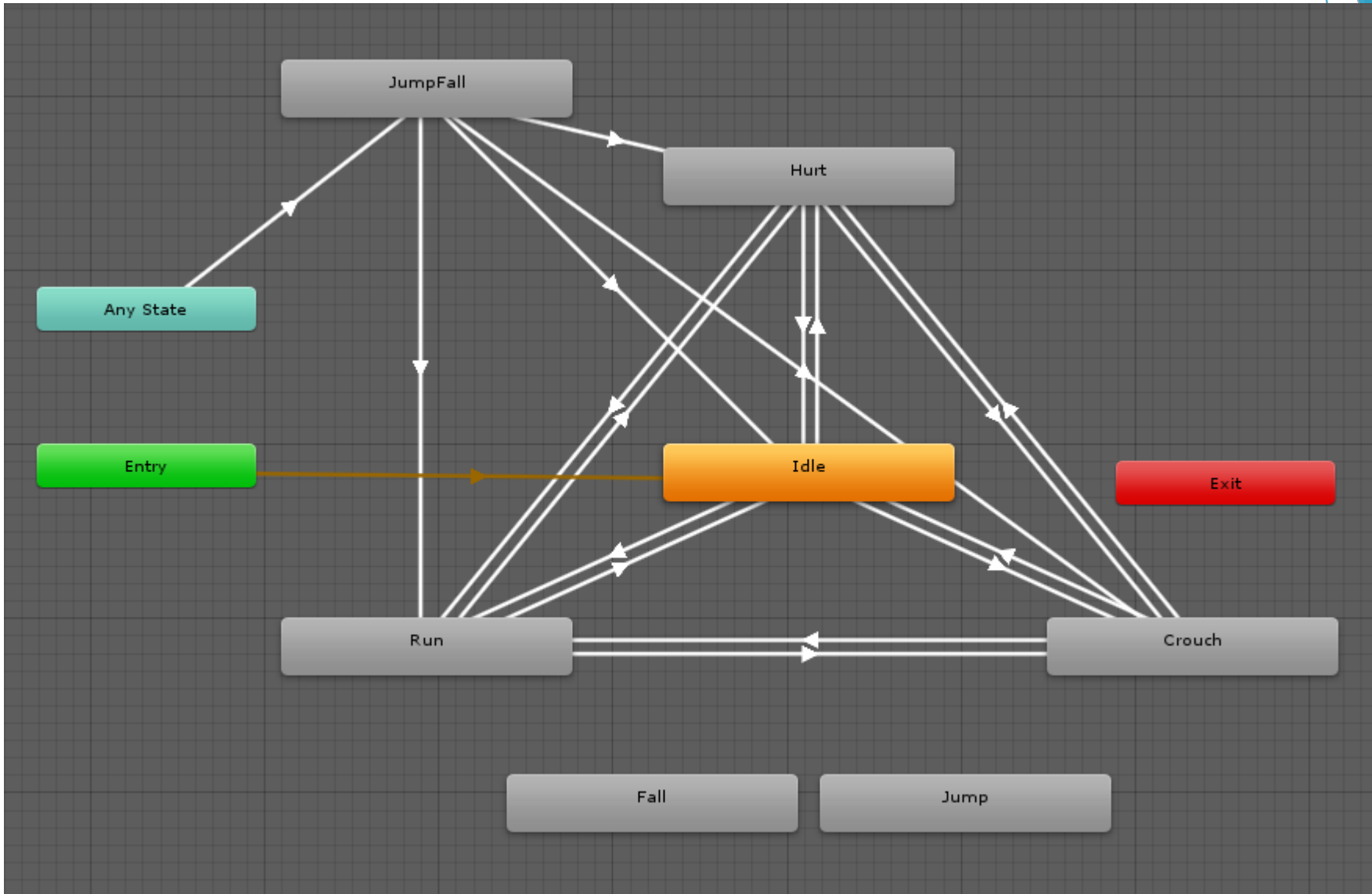
Inspector - Unity



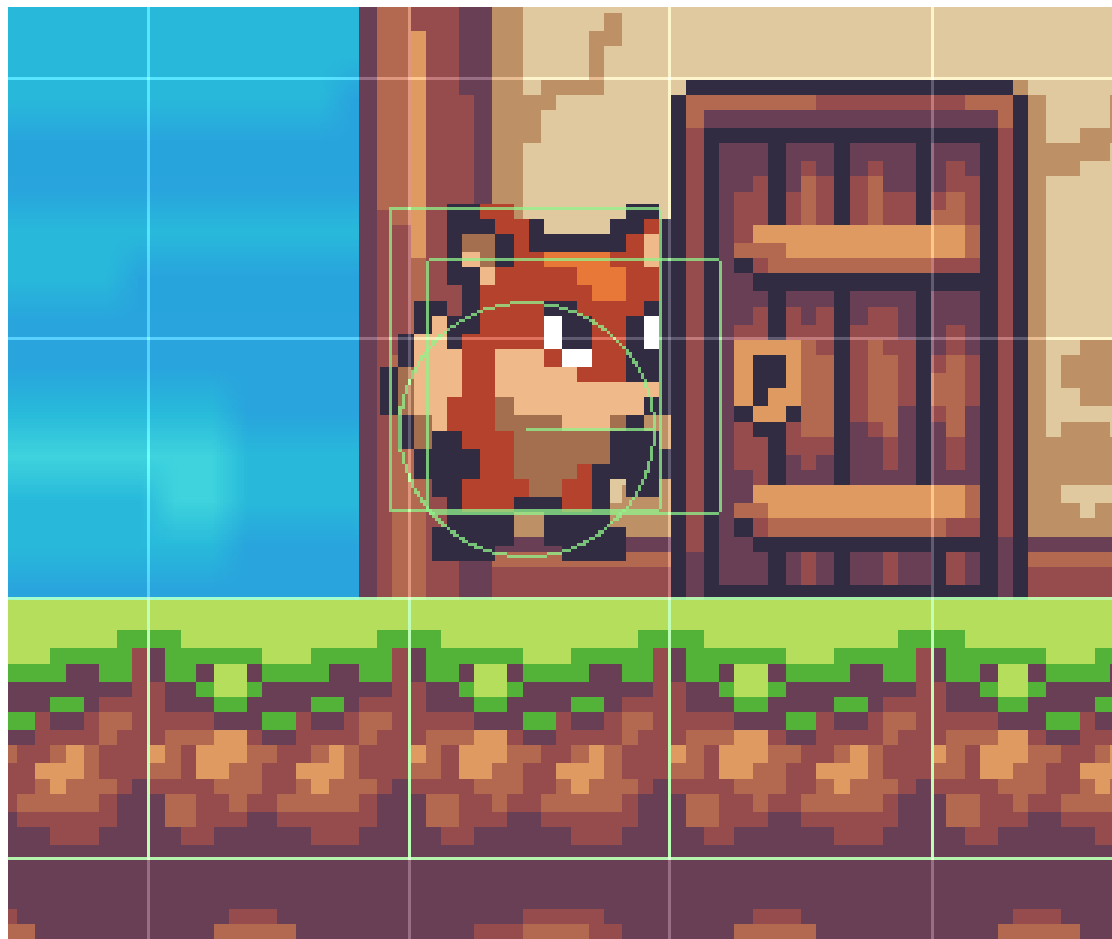
Scene e Hierarchy - Unity



Animator - Unity



Colisores - Unity



Movimentação



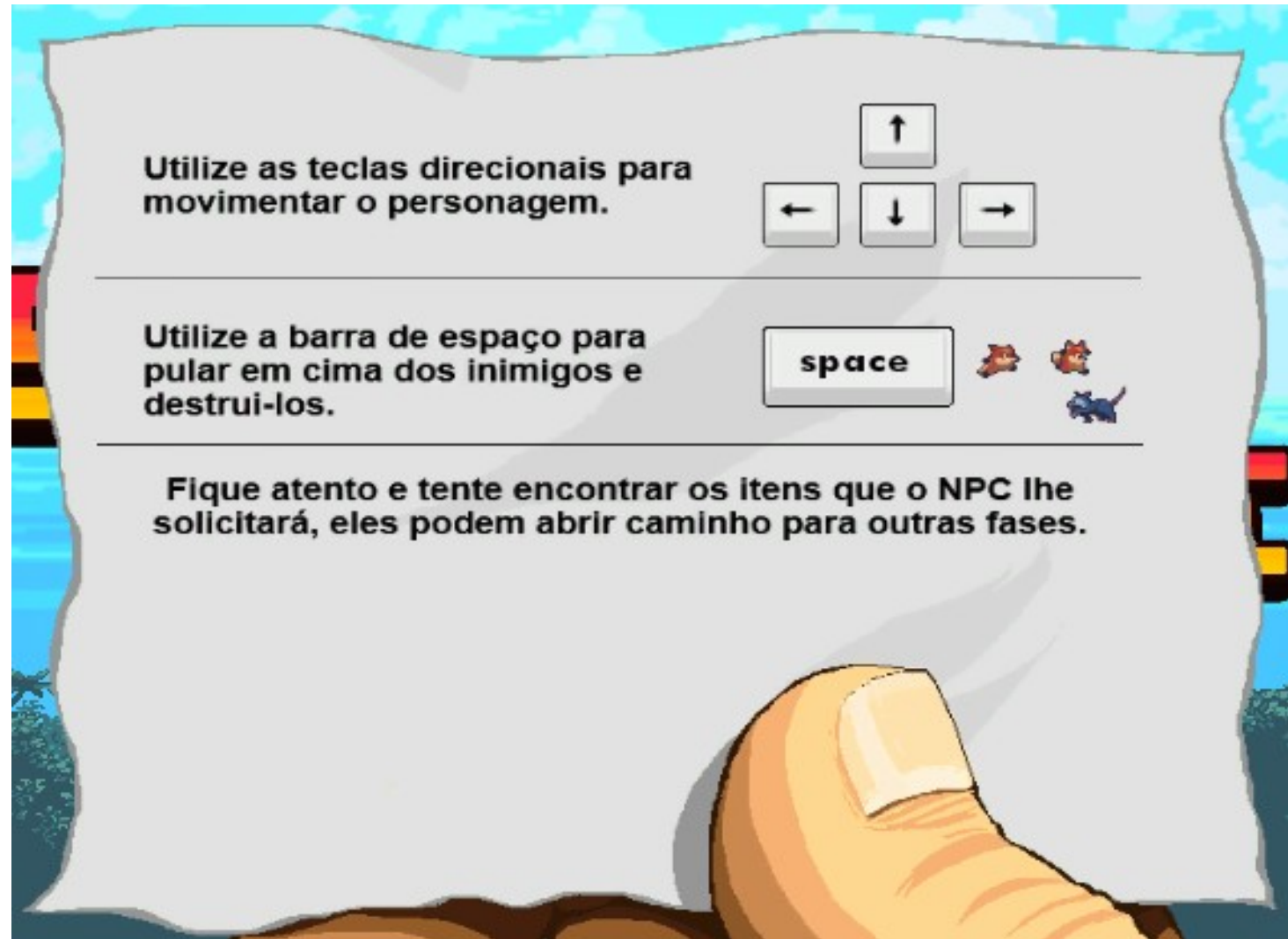
Testes

Nº do Teste	Tipo do Teste	Objetivo do Teste	Resultado Esperado
RF0001	Funcional	Validar movimentação do personagem.	Movimentar o personagem livremente conforme os comandos do teclado.
RF0002	Funcional	Validar colisão do personagem e dos inimigos com outros objetos.	Colidir com obstáculos.
RF0003	Funcional	Validar eventos de morte.	Executar a animação de morte quando for preciso.
RF0003	Funcional	Validar interação com outros objetos.	Interagir e executar a ação programada para aquele objeto.

Implantação

- Configuração do ambiente de hospedagem.
- Disponibilização do software para download
- Manual do Usuário.

Manual do Usuário



Análise e Resultados

- Plano de teste.
- Escolha da ferramenta.
- Estudo das funcionalidades da ferramenta.

Considerações Finais

- Interesse Profissional.
- Experiência adquirida.
- Resultado final.

Referências

AMOROSO, Danilo. **A história dos Videogames Games: Do osciloscópio aos Gráficos 3D.** Disponível em: <https://www.tecmundo.com.br/xbox-360/3236-a-historia-dos-video-games-do-osciloscopio-aos-graficos-3d.htm>. Acessado em: 30 de Março de 2019.

História do Videogame. Disponível em: <https://www.historiadetudo.com/videogame>. Acessado em: 30 de Março de 2019.

MICROSOFT. Disponível em: <https://visualstudio.microsoft.com/pt-br/?rr=https%3A%2F%2Fwww.google.com%2F>. Acessado em: 13 de Abril de 2019.

Unity. Disponível em: <https://unity.com/pt> . Acessado em: 21 de Março de 2019.

Unity Asset Store. Disponível em: https://assetstore.unity.com/?_ga=2.89442331.1440210711.1559235970-871820549.1559235970 . Acessado em: 21 de Março de 2019.

Atari. Disponível em : <https://www.atari.com/about-us/>. Acessado em: 22 de Junho de 2019.



GitHub

Disponível em:

<https://github.com/joao-ranieri/2D-Game>

Obrigado!

João Paulo Souza Ranieri
Joao.s.ranieri@gmail.com

